

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-36381

(43)公開日 平成7年(1995)2月7日

(51)Int.Cl. ⁶	識別記号	序内整理番号	F I	技術表示箇所
G 0 9 B 29/10	A			
G 0 1 C 21/00	N			
G 0 6 T 1/00				
		8125-5L	G 0 6 F 15/ 62	3 3 5
		8837-5L	15/ 70	3 6 5
		審査請求 未請求 請求項の数 2 O L (全 13 頁) 最終頁に続く		

(21)出願番号	特願平5-178135	(71)出願人	000002130 住友電気工業株式会社 大阪府大阪市中央区北浜四丁目5番33号
(22)出願日	平成5年(1993)7月19日	(72)発明者	天目 健二 大阪市此花区島屋一丁目1番3号 住友電 気工業株式会社大阪製作所内

(74)代理人 弁理士 亀井 弘勝 (外1名)

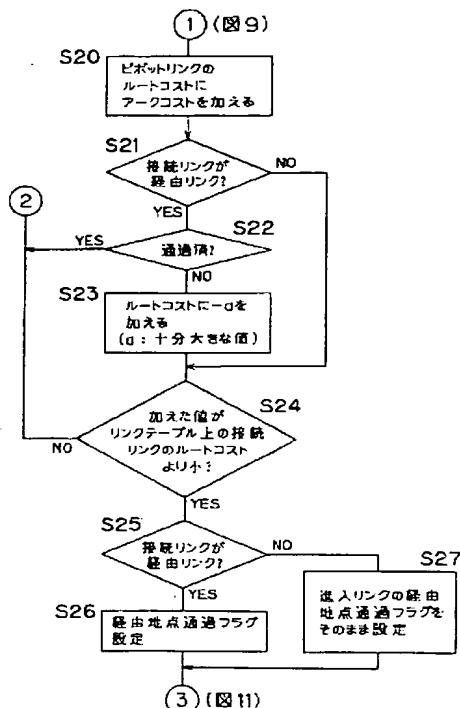
(54)【発明の名称】 経路計算方法

(57)【要約】

【目的】経由地点がある場合、出発地点から目的地点に至る経由地点を含むリンクのツリーを用いて1回で経路計算する。

【構成】経路計算するときに、経由地点を含むリンクを設定し、そのリンクについてルートコスト値に十分大きな負の値を加えるとともに、当該リンク及び以後これに接続される経路に当該経由地点通過済みのステータスを与え、後に探索されたリンクが経由地点に戻ったときに、当該経由地点通過済みのステータスを有していれば、その接続リンクについての処理を打切る。

【効果】経由地点を必ず通過し、しかも1回のみ通過する最適経路が、1回の経路計算により得られる。



【特許請求の範囲】

【請求項1】経路ネットワークを構成する各リンクのコスト及びリンク相互の接続関係を記憶した経路ネットワークメモリと、

リンクごとに、リンクコストと、このリンクの次に接続される1つ又は複数の接続リンクと、このリンクの前に接続される計算済の経路上のリンクと、前記計算済の経路のルートコストとを記入する欄を有する第1のテーブルと、

現在接続リンクを探索しているリンクを記入する欄を有する第2のテーブルとを用いて、

(a) 出発地点、経由地点及び目的地点の設定に応じて、経路ネットワークメモリから一定範囲の経路ネットワークデータを読み出し、

(b) 読み出された経路ネットワークデータに基づいて、経路ネットワークを構成する各リンクごとのリンクコスト、接続リンクを特定するとともに、各リンクのルートコストの初期値を十分大きな値に設定して、前記第1のテーブルの該当欄に書き込み、

(c) 出発地点を含むリンクを第2のテーブルに書き込み、この第2のテーブルに書き込まれたリンクに対応する第1のテーブルのルートコスト欄を0に設定し、

(d) 第2のテーブルに記入されたリンクについて、第1のテーブルを参照して接続リンクを探索し、探索された接続リンクのリンクコストを前記第2のテーブルに記入されたリンクコストに加え、この加えた値を、第1のテーブルに記入された当該接続リンクのルートコストと比較し、

(e) 比較の結果、加えられた値が大きければ、その接続リンクについての処理を打切り、小さければその値を第1のテーブルの当該接続リンクのルートコストに書き換えるとともに、前記第2のテーブルに記入されたリンクをこの接続リンクに到る計算済の経路上のリンクとして書き込み、

(f) 当該接続リンクを第2のテーブルに新たに記入して、前記(d)、(e)の処理を繰り返し、

(g) 第2のテーブルに記入された全てのリンクについて接続リンクの探索が終了したときに、目的地点を含むリンクを第1のテーブルから探し出し、そのリンクに到る計算済の経路上のリンクを順に辿っていくことにより、最適経路を決定する経路計算方法において、

前記手順(d)において、探索された接続リンクが経由地点を含むリンクであるときに、前記値に十分大きな負の値を加えて比較するとともに、当該接続リンク及び以後探索されるこれに接続されるリンクに当該経由地点通過済みのステータスを与え、

前記手順(d)において、探索された接続リンクが経由地点を含むリンクであるときに、この接続リンクが当該経由地点通過済みのステータスを有していれば、その接続リンクについての探索処理を打切ることを特徴とする経

路計算方法。

【請求項2】経由地点の数が複数個ある場合、前記経由地点通過済みのステータスを、各経由地点に応じて区別して設定し、第1のテーブルの「前に接続される計算済の経路上のリンク」を、各経由地点通過・非通過の組み合せに応じて複数通り設定することを特徴とする請求項1記載の経路計算方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、運転者による目的地点の設定に応じて、道路地図メモリから出発地点（車両の現在位置でもよい）と経由地点と目的地点とを含む範囲の経路ネットワークデータを読み出し、この経路ネットワークデータに基づいて経由地点を通る最適経路を計算して運転者に示すことができる経路計算方法に関するものである。

【0002】

【従来の技術】従来より画面上に車両の位置方位等を表示し、見知らぬ土地や夜間等における走行の便宜を図るために開発されたナビゲーション装置が知られている。前記ナビゲーション装置は、ディスプレイ、方位センサ、距離センサ、道路地図メモリ、コンピュータを車両に搭載し、方位センサから入力される方位データ、距離センサから入力される走行距離データ、及び道路地図メモリに格納されている道路パターンとの一致に基づいて車両位置を検出し、この車両位置を道路地図とともにディスプレイに表示するものである。

【0003】この場合、出発地点から目的地点に至る走行経路の選択をするために、運転者による目的地点設定入力に応じて現在の出発地点から目的地点までの経路をダイクストラ法を用いてコンピュータにより自動的に計算する方法が提案されている（特開平5-53504号公報参照）。この方法は計算の対象となる道路を幾つも区切って、区切った点をノードとし、ノードとノードとを結ぶ経路をリンクとし、出発地点（目的地点でもよい）に最も近いノード又はリンクを始点とし、目的地点（出発地点でもよい）に最も近いノード又はリンクを終点とし、始点から終点に至るリンクのツリーを探索し、ツリーを構成する全ての経路のリンクコストを順次加算して、目的地点又は出発地点に到達する最もリンクコストの少ない経路のみを選択する方法である。

【0004】この方法で経路を計算し、経路に沿って走行していくば確実に目的地点に到達するので、道を知らない運転者にとって便利である。

【0005】

【発明が解決しようとする課題】しかしながら、運転者が目的地点に到達する前に立ち寄りたい地点、すなわち経由地点がある場合、出発地点から目的地点に至るリンクのツリーを用いて1回で経路計算するという訳にはいかない。この場合、出発地点から経由地点までの最適経

路を計算し、次に経由地点から目的地点までの最適経路を計算し、最後に両経路をつなぎ合わせるという方法が普通とされる。

【0006】しかし、複数回経路計算をしなければならず、処理時間がそれだけ長くなる。という問題がある。前記のリンクのツリーは、通常、膨大なデータ量になるため経路計算時間は経由地点がなくとも長くなりがちであるので、それ以上に処理時間が長くなるというのは、問題がある。本発明は、前記の問題に鑑みてなされたもので、経由地点が1つ又は複数設定されている場合であっても1回で経路計算することができ、計算時間が長くならない経路計算方法を提供することを目的とする。

【0007】

【課題を解決するための手段】前記の目的を達成するための請求項1記載の発明の構成を説明する前に、経路計算の前提としたいわゆるポテンシャル法について説明する。このポテンシャル法は、請求項1の「…において」の部分に記載したように、出発地リンクからスタートして、それに接続するリンクを探索し、探索された接続リンクのルートコストが前に設定された同じリンクのルートコストより小さくなれば、その値を当該接続リンクのルートコストに書き換えるとともに、経路をこの新しいルートに変更し、次にこの接続リンクから出発して以上の手順を繰り返し、すべてのリンクについて探索が終了すれば、最適経路を決定する方法である。

【0008】本発明では、このポテンシャル法を実行する場合に、経由地点を含むリンクを設定し、探索された接続リンクが経由地点を含むリンクであるときに、ルートコスト値に十分大きな負の値を加えるとともに、当該接続リンク及び以後これに接続されるリンクに当該経由地点通過済みのステータスを与え、後に、探索された接続リンクが経由地点を含むリンクであるときに、当該経由地点通過済みのステータスを有していれば、その接続リンクについての処理を打切るものである。

【0009】請求項2記載の発明によれば、経由地点の数が複数個ある場合、前記経由地点通過済みのステータスを、各経由地点に応じて区別して設定し、第1のテーブルの「前に接続される計算済の経路上のリンク」を、各経由地点通過・非通過の組み合わせに応じて複数通り設定している。

【0010】

【作用】この経路計算方法によれば、探索された接続リンクが経由地点を含むリンクであるときに、前記値は十分大きな負の値になるので、請求項1の(e)の手順において、「比較の結果、加えられた値が——小さければその値を第1のテーブルの当該接続リンクのルートコストに書き換えるとともに、第2のテーブルに記入されたリンクをこの接続リンクに到る計算済の経路上のリンクとして書き込む処理を受けることになる。その結果、この経由地点を含むリンクに到る経路が前の経路から変更さ

れることになる。すなわち、最適経路は、必ず経由地点を含むリンクを通るようになる。

【0011】また、当該接続リンク及び以後これに接続されるリンクに当該経由地点通過済みのステータスを与えるので、後に、探索された接続リンクがもう一度経由地点を含むリンクに戻ってきたとき（閉ループを作ったとき）に、当該経由地点通過済みのステータスを有していれば、その接続リンクについての処理を打切る。したがって、経由地点の通過が2回以上になることはない。

【0012】また、請求項2記載の発明によれば、経由地点の数が複数個ある場合、前記経由地点通過済みのステータスを、各経由地点に応じて区別して設定し、第1のテーブルの「前に接続される計算済の経路上のリンク」を、各経由地点通過・非通過の組み合わせに応じて複数通り設定しているので、各経由地点を1回ずつ通る最適経路を1度の計算で求めることができる。

【0013】

【実施例】以下本発明の実施例を示す添付図面に基づいて詳細に説明する。本発明の経路計算方法を実施するナビゲーション装置は、図2に示すように、方位センサ11と、距離センサ12と、シフトセンサ13と、道路地図データを格納している道路地図メモリ14A及び経路ネットワークデータを格納した経路ネットワークメモリ14Bから記憶データを読出すメモリドライブ15と、距離センサ12により検出される走行距離及び方位センサ11により検出される走行方向変化量をそれぞれ積算し、この積算データとメモリドライブ15により読出した道路地図データとの比較に基いて車両位置を検出するロケータ16と、所定範囲の道路地図の読み出し、経路ネットワークデータを利用した最適経路の計算、車両の誘導をするための表示用データの作成、音声出力装置の制御、及びロケータ16の制御等の種々の制御を行うコントローラ17と、初期データ等を設定するためのタッチキー20と、ディスプレイ19とを有する。

【0014】さらに詳細に説明すれば方位センサ11は、車両の走行に伴なう方位の変化を検出するものであり、地磁気センサ、ジャイロ等を使用することが可能である。距離センサ12は、車両の速度、あるいは、車輪の回転数等に基づいて走行距離を検出するものであり、車輪速センサ、車速センサ等が使用可能である。ロケータ16は、距離センサ12により検出される距離データ、及び方位センサ11により検出される方位変化データをそれぞれ積算して走行軌跡データを算出し、走行軌跡データと道路地図メモリ14Aに格納されている道路のパターンとの比較（いわゆる地図マッチング法、特開昭64-53112号公報参照）に基いて車両位置を検出する。なお、位置検出の精度をあげるためにビーコン受信機やGPS受信機を付加してもよい。

【0015】ディスプレイ19は、CRT、液晶パネル等の画面上にメニュー画面の表示、道路地図の表示、車

両の現在位置と方位の表示、最適経路の表示、目的地点や目印となる地点の表示、目的地点までの方位と距離の表示、複合交差点の形状拡大画面のウィンドウ表示等を行うものである。また、ディスプレイ19には、透明のタッチキー20が取付けられている。タッチキー20は、初期設定メニュー画面上で最適経路の選定基準（最短時間経路、最短距離経路等）、地図の倍率、目的地点等を入力するものである。すなわち、経路誘導装置と運転者との対話を仲介している。

【0016】各メモリ14A、14Bは、大容量記憶媒体であるCD-ROM、ICメモリカード、磁気テープ等のメモリ等から構成されている。道路地図メモリ14Aは、道路地図（高速自動車国道、都市高速道路、一般国道、主要地方道、一般都道府県道、指定都市の一般市道、その他の生活道路を含む）をメッシュ状に分割し、各メッシュ単位でノードとリンクとを組み合わせたデータを記憶している。その他、鉄道、川、地名欄、有名施設、運転者が予め登録した地点、等高線等の表示用の背景データを含んでいてもよい。

【0017】前記メッシュは、日本道路地図を経度差1度、緯度差40分で分割し縦横の距離を約80Km×80Kmとした第1次メッシュと、この第1次メッシュを縦横8等分し縦横の距離を約10Km×10Kmとする第2次メッシュとの二重構造を持っている。ここでノードとは、一般に、道路の交差点や折曲点を特定するための座標位置のことであり、交差点を表わすノードを交差点ノード、道路の折曲点（交差点を除く）を表わすノードを補間点ノードということがある。

【0018】各ノードを繋いだものがリンクである。リンクデータはリンク番号、リンクの始点ノード及び終点ノードのアドレス、リンクの距離、リンクを通過する方向、その方向における所要時間、道路種別、道路幅、一方通行や有料道路等の通行規制データ等からなる。このように、リンクデータの中にリンクの始点ノード及び終点ノードのアドレスが入っていることから、リンクのみによっても地点を特定できる。なお、リンクによってリンクの始点を特定する場合そのリンクを「退出リンク」といい、リンクの終点を特定する場合そのリンクを「進入リンク」という。さらに、リンクデータにはリンクを通過する方向が入っているので、1つのリンクを特定することにより、車両の進行方向も特定することができる。図3は十字路を特定する4つの退出リンクを、図4は十字路を特定する4つの進入リンクを例示している。以下の実施例では、地点を特定するのに主に退出リンクを用いるものとする。

【0019】コントローラ17の詳細を図5に示す。コントローラ17は、メモリドライブ15を通して道路地図メモリ14A及び経路ネットワークメモリ14Bから必要なデータを得るメモリ制御部21、音声出力装置に電子的に記録された音声を発声させる音声制御部25、

ディスプレイ19に必要な画像を表示させる表示制御部22、タッチキー20で設定された入力情報を処理する入力処理部23、ロケータ16の算出した車両位置をデータとして取り込む車両位置認識処理部24、ポテンシヤル法により目的地点から出発地点までの最適経路を計算する経路計算処理部26が接続される。

【0020】経路計算処理部26は、経路ネットワークメモリ14Bから取り出された出発地点、経由地点及び目的地点を含む一定の範囲のリンク情報に基づいて、リンクテーブルと、各リンクの接続リンクを探索するためのアクトテーブルとを作成し、これらのテーブルに基づいて最適経路を計算するCPU26aと、リンクテーブルとアクトテーブルとピボットテーブルとを蓄える主メモリ26cと、アクトテーブルやリンクテーブル、ピボットテーブルの内容の一部をコピーし蓄えるキャッシュメモリ26bとを備えている。なお、リンクテーブルとアクトテーブルとが、請求項1の「第1のテーブル」に相当し、ピボットテーブルが「第2のテーブル」に相当するものである。

【0021】リンクテーブルは、経路ネットワークメモリ14Bから取り出された出発地点、経由地点及び目的地点を含む一定の範囲のリンクについて、シーケンス番号、リンク番号、メッシュの境界リンクに対して設定される隣接メッシュポインタ、アクトテーブルへのポインタ、リンクコスト、ルートコスト、そのリンクに到達する前の経路上のリンク（以下「前リンク」という）、当該リンクまでの経路が経由地点を通過しているかどうかを示す経由地点通過フラグ、並びに始端ノードの番号と終端ノードの番号を記憶している。

【0022】前記シーケンス番号は0から昇順に設定された整数、リンク番号は往復リンクのペアごとに付与される番号である。ルートコストは経路計算の途中経過を記録した当該リンクへの到達コストである。さらに正確には、あるリンクのルートコストとは、出発値から当該リンクまでつながった経路上の各リンクのアクトコストを総和したものとなる（図6参照）。ここで、アクトというのは、リンクの始端ノードの直後から次のリンクの始端ノードの直後までをいう。アクトコストというのは、当該リンクのリンクコストと当該リンクから退出するための右左折又は直進コストの和である。例えば、進入禁止の場合、リンクコストは有限であっても直進コストは無限大となるので、アクトコストも無限大となる。信号がある場合、アクトコストは、リンクコストに右左折又は直進時の平均的な信号待ち時間を考慮したもの足したコストとなる。

【0023】前リンクは、経由地点数をNとすると、経由地点の通過・非通過の組み合わせに応じて 2^N 通りの数だけある（詳細は後述する）。アクトテーブルは、シーケンス番号、同じリンクから出るアクトのうち最後のアクトかどうかを示すアクト終端識別子、当該アクトの

接続リンクのリンクテーブルへのポインタ（リンクテーブルのシーケンス番号で示す）及びアークコストを記憶している。

【0024】アーク終端識別子は、例えば1つのリンクから3つのアークが出る場合を想定すると、1番目と2番目とのアークについてはアーク終端識別子は0が割り付けられ、3番目のアークについて1が割り付けられる。したがって、アークテーブルを用いて1つのリンクに接続されるリンクを探索するとき、アーク終端識別子が0のうちは、他の接続リンクが存在することが分かる。アーク終端識別子が1になれば、もうこれ以上接続リンクはないことが分かる。

【0025】さらに具体的な経路ネットワークを想定して説明する。図7は、出発地点と目的地点とを含む経路計算の対象となる範囲内のノードをA, B, C, D, *

リンクテーブル

シーケンス番号	リンク番号	隣接メッシュポインタ	アークテーブルへのポインタ	リンクコスト	ルートコスト	前リンク	経由地点通過フラグ	始端ノード	終端ノード
0	1	—	0	30				A	B
1	2	—	2	25				A	C
2	1	—	4	30				B	A
3	3	—	5	40				B	C
4	4	—	7	35				B	E
5	2	—	8	20				C	A
6	3	—	9	35				C	B
7	5	—	11	15				C	D
8	5	—	12	∞				D	C
9	6	—	14	10				D	E
10	4	—	15	25				E	B
11	6	—	17	∞				E	D
12	7	—	—	—				E	F

【0028】表2は、アークテーブルであり、シーケンス番号、アーク終端識別子、リンクテーブルへのポインタ及びアークコストの欄を備えている。

*E, Fとし、Aを出発地点ノードとし、Eを目的地点ノードとしたリンク経路図を示す。全リンク数は、A→B, A→C, B→A, B→C, B→E, C→A, C→B, C→D, D→C, D→E, E→B, E→D, E→Fの13個である。

【0026】記憶されたテーブルの構造をそれぞれ表1、表2に示す。表1はリンクテーブルであり、13個のリンクに対してシーケンス番号、リンク番号、隣接メッシュポインタ、アークテーブルへのポインタ、リンクコスト、ルートコスト、前リンク、経由地点通過フラグ、始端ノードの番号と終端ノードの番号の欄を備えている。

【0027】

【表1】

【0029】

【表2】

アークテーブル

シーケンス番号	アーク終端識別子	リンクテーブルへのポインタ	アークコスト	備考
0	0	4		B→E
1	1	3		B→C
2	0	6		C→B
3	1	7		C→D
4	1	1		A→C
5	0	5		C→A
6	1	7		C→D
7	1	12		E→F
8	1	0		A→B
9	0	2		B→A
10	1	4		B→E
11	1	9		D→E
12	0	5		C→A
13	1	6		C→B
14	1	12		E→F
15	0	2		B→A
16	1	3		B→C
17	1	8		D→C

【0030】ピボットテーブルは、リンクテーブルに記憶されている多数のリンクのうち、現在計算に必要なリンクのシーケンス番号等を取り出して記憶している先入れ先だし型のテーブルである。次に、リンクテーブル、アークテーブル及びピボットテーブルを利用した経路探索手法を説明する。

【0031】図7を参照してリンクA→Bから出発するとする。目的地点はノードEであるとする。リンクテーブルのリンクA→Bに対応するシーケンス番号0の欄を見ると、アークテーブルへのポインタは0を示しているので、アークテーブルのシーケンス番号0の欄を見る。リンクテーブルへのポインタは4となっている。そこで、リンクテーブルのシーケンス番号4の欄を見ると、シーケンス番号4のリンクは始端ノードBと終端ノードEによって特定されるリンクB→Eである。したがって、リンクA→BにリンクB→Eが接続されることが分かる。アークテーブルのアーク終端識別子は0となっているので、リンクA→Bに続くアークが他にも存在する。そこで、アークテーブルを下に辿って、シーケンス番号1の欄を見る。リンクテーブルへのポインタは3となっているので、リンクテーブルのシーケンス番号3の欄を見ると、リンクB→Cである。したがって、リンクA→BにリンクB→Cが接続されることが分かる。ここで、アーク終端識別子は1となっているので、リンクA

→Bに接続するリンクの探索はこれで終了となる。

【0032】次に、リンクB→Eに接続されるリンクを探索する。リンクテーブルのシーケンス番号4の欄を見ると、アークテーブルへのポインタは7となっている。そこで、アークテーブルのシーケンス番号7の欄を見ると、リンクはリンクE→Fである。アークテーブルのシーケンス番号7の欄のアーク終端識別子は1となっているので、リンクB→Eに接続されるアークはE→Fのみである。

【0033】次に、リンクB→Cに接続されるリンクを探索する。リンクテーブルのシーケンス番号3の欄を見ると、アークテーブルへのポインタは5となっている。そこで、アークテーブルのシーケンス番号5の欄を見ると、リンクはリンクC→Aである。アークテーブルのシーケンス番号5の欄のアーク終端識別子は0となっているので、リンクB→Cに接続されるアークはC→Aのみでない。そこで、アークテーブルの次の欄6を見ると、リンクテーブルへのポインタは7となっている。そこで、リンクテーブルのシーケンス番号7の欄を見ると、リンクC→Dである。アーク終端識別子は1となっているので、これ以外に接続リンクはないことになる。

【0034】以上のようにして接続リンクがツリーのように次々と求められる。求められたツリーは図8に示すようなものである。ツリーは無限に広がるが、後述する

ように処理に工夫を加えることにより、ツリーの途中で計算を打ち切るようにしている。以上のようにしてリンクを次々と探索していく方法を説明したが、経路計算処理では、経路の探索に加えて、アークコストを積算していく、最もコストの少ない最適な経路を計算する必要がある。

【0035】そこで最適経路計算の仕方を、フローチャート(図9-12)を用いて詳細に説明する。まず、CPU26aは、出発地点、目的地点を含む一定の範囲のリンクを経路ネットワークメモリ14Bから取り出し、所定のプログラムに基づいて、リンクテーブルを作成する。このとき、リンクテーブルのルートコストの値はすべて無限大(実際には、予想されるルートコストより2桁程度大きな値に設定すれば十分である。)、前リンクなし、経由地点通過フラグ0として初期設定する。

【0036】以下、図7の経路ネットワークを想定し、出発地リンクをA→B、目的地リンクをE→Fとして最適経路を計算する。まず、経由地点を指定しない場合の計算方法を図9、10を用いて説明する。

(1) 経由地点の指定のない場合

CPU26aは、出発地リンク(初期リンクという)A→Bを取り出し、この初期リンクA→Bをピボットテーブルに登録する(ステップS1)。ピボットテーブルに登録されたリンクを、以下「ピボットリンク」という。このピボットリンクに対応するルートコストをリンクテーブル上で0にしておく。

【0037】CPU26aは、ピボットテーブルを参照し(ステップS2)、未処理ピボットリンクがあるかどうかを判定する(ステップS3)。最初は何の処理もしていないので、未処理ピボットリンクすなわち初期リンクA→Bが存在する。これを未処理ピボットリンクとして取り出す(ステップS4)。CPU26aは、全アークの探索が終了したかどうかを判断する(ステップS5)。探索が終了していないければ、ピボットリンクA→Bにつながるリンクを探索するため、ピボットリンクA→Bに対応するリンクテーブルを参照する(ステップS6)。ステップS7に進み、リンクテーブルの当該欄のアークコストとリンクテーブルへのポインタを参照する。その結果、前述したように、リンクB→Eが探し出される。アークコストは、リンクA→Bのリンクコストに交差点Bでの直進コストを加えたものになる。そして、図10のステップS8に移り、ピボットリンクA→Bのルートコスト(0になっている)に前記アークコストを加える。そして、このコストをリンクテーブルに書き込まれた接続リンクB→Eのルートコスト(初期値は無限大となっている)より小さいかどうか判定する(ステップS9)。

【0038】小さければ、ステップS10に進み、接続リンクB→Eのルートコストをリンクテーブル上で書き換える。そして接続リンクB→Eの前リンクとして、リ

ンクA→Bを設定する(ステップS11)。さらにこの接続リンクB→Eをピボットテーブルに登録する(ステップS12)。

【0039】その後、ステップS5に戻り、当該リンクA→Bにつながる他のリンクを探索するため、リンクテーブルの次の欄を参照する(ステップS6)。ステップS7に進み、リンクテーブルの当該欄のアークコストとリンクテーブルへのポインタが参照される。その結果、リンクB→Cが探し出される。アークコストは、リンクA→Bのリンクコストに交差点Bでの右折コストを加えたものになる。そして、図10のステップS8に移り、ピボットテーブルのリンクA→Bのルートコスト(0になっている)に前記アークコストを加える。そして、このルートコストをリンクテーブルに書き込まれた接続リンクB→Cのルートコスト(初期値は無限大になっている)より小さいかどうか判定する(ステップS9)。

【0040】小さければ、ステップS10に進み、接続リンクB→Cのルートコストをリンクテーブル上で書き換える。そして接続リンクB→Cの前リンクとして、リンクA→Bを設定する(ステップS11)。さらにこの接続リンクB→Cをピボットテーブルに登録する(ステップS12)。その後、ステップS5に戻り、ピボットリンクA→Bに対応するリンクテーブルのアーク終端識別子を参照すると、リンクA→Bにつながる他のリンクは存在しないので、全アークの探索が終了したことが分かる。ステップS31に進み、ピボットリンクA→Bに処理済フラグを立てる。そしてピボットテーブルを参照し(ステップS2)、未処理ピボットリンクがあるかどうかを判定する(ステップS3)。この段階では、リンクB→E、B→Cがピボットテーブルに未処理として登録されている。そこで、例えば未処理ピボットリンクB→Eを取り出し、上と同じ処理をする。すなわち、接続リンクであるリンクE→Fを探し出し、ピボットリンクB→Eのルートコストに、アークコスト(リンクB→Eのリンクコストに交差点Eでの直進コストを加えたもの)を加える。そしてこれをリンクE→Fのルートコストとおき、リンクテーブル上のリンクE→Fのルートコストを書き換える。これで目的地Eに到る1つのルートコストが求まることになる。

【0041】ピボットリンクB→Cについても同じ処理をして、リンクB→CにつながるリンクC→A、C→Dを探索し、それらのルートコストを書き換える。なお、リンクC→Aを探索した後、リンクC→Aに接続するリンクをリンクテーブルから拾うと、初期リンクA→Bに戻ってしまい、経路が閉ループとなって探索が無限に続くおそれがあるが、この場合はステップS9において、リンクA→Bの新しいルートコストを、リンクA→Bのルートコストとして登録されている値(初期値=0)と比較する。新しいルートコストは、経由地の指定がない場合は負にはならないので、常に、

(新しいルートコスト) > (前のルートコスト = 0) という関係が成立し、ステップS 5に戻る。したがつて、無限ループに陥るおそれはない。

【0042】以上ようなことを続けると、最後に、リンクE→Fのルートコストが再び求められる。このとき、ステップS 9において、新しいルートコストを前にリンクテーブル上にリンクE→Fのルートコストとして登録された値と比較する。リンクE→Fの新しいルートコストが前のルートコストより大きければ、リンクテーブルを書き換えることなく、ステップS 5に戻る。

【0043】そして、ステップS 5、S 31、S 2を経由してステップS 3に至り未処理ピボットリンクがあるかどうかを判定する。ピボットリンクがすべて処理済みとなると、経路の探索は終了したことになる。CPU 26aはリンクテーブルの目的地リンクに注目し、目的地リンクの前リンクを辿っていくと、出発値に到達する。その経路が最適経路になる(ステップS 42)。また、目的地リンクのルートコストを参照すると、そのルートコストが出発値から目的地までの最適経路ルートコストとなる。

【0044】次に、経由地点を指定する場合の計算方法を図9、11、12を用いて説明する。

(2) 経由地点が1つある場合

経由地リンクをリンクB→Cとする。CPU 26aは、初期リンクA→Bをピボットテーブルに登録し(ステップS 1)、このピボットリンクに対応するルートコストをリンクテーブル上で0にしておく。

【0045】CPU 26aは、ピボットテーブルを参照し(ステップS 2)、初期リンクA→Bを未処理ピボットリンクとして取り出す(ステップS 4)。CPU 26aは、アクトテーブルを参照し、当該リンクA→Bにつながるリンクを探索する(ステップS 6)。その結果、まず、リンクB→Eが探し出される。ステップS 7に進み、アクトテーブルの当該欄のアクトコストを参照する。そして、図1のステップS 20に移り、ピボットリンクA→Bのルートコスト(0になっている)に前記アクトコストを加える。次に、接続リンクB→Eが経由地リンクB→Cかどうかを判定する(ステップS 21)。経由地リンクB→Cでなければ、ステップS 24に進み、このルートコストをリンクテーブルに書き込まれた接続リンクB→Eのルートコスト(初期値は無限大)より小さいかどうか判定する。

【0046】小さければ、ステップS 25に進み、接続リンクB→Eが経由地リンクB→Cかどうかを判定する。経由地リンクB→Cでなければ、接続リンクB→Eの経由地点通過フラグとして、進入リンクA→Bの経由地点通過フラグの値(初期値0)をそのまま設定する(ステップS 27)。その後、図1のステップS 28に進み、接続リンクB→Eのルートコストを書き換え、接続リンクB→Eの前リンクとして、リンクA→Bを設

定する(ステップS 29)。さらにこの接続リンクB→Eをピボットテーブルに登録する(ステップS 30)。

【0047】その後、ステップS 5に戻り、当該リンクA→Bにつながる他のリンクを探索する(ステップS 6)。その結果、リンクB→Cが探し出される。ステップS 7に進み、アクトテーブルの当該欄のアクトコストとリンクテーブルへのポインタが参照される。アクトコストは、リンクA→Bのリンクコストに交差点Bでの右折コストを加えたものになる。そして、図1のステップS 20に移り、ピボットテーブルのリンクA→Bのルートコスト(0になっている)に前記アクトコストを加える。

【0048】ステップS 21に移り、接続リンクB→Cが経由地リンクB→Cかどうかが判定される。経由地リンクB→Cであれば、ステップS 22に進み、この接続リンクB→Cは経由地点通過済みかどうかが判定される。この判定は、経由地点を2度通過しないように行われるもので、経由地点通過フラグが立っているかどうかにより判定される。通過済みであれば、この接続リンクB→Eの処理は打切り、ステップS 5にもどり、他の接続リンクを探す。経由地点が未通過であれば、ステップS 23に進み、ピボットリンクA→Bのルートコストに前記アクトコストを加えた値(ステップS 20で求めたもの)に-aを加える。aは出発地点から目的地点まで予想されるルートコストよりも2桁程度大きな値であればよいが、もっと大きな値でもよい。これにより、経由地点通過時のルートコストを大きく下げる、経由地点の通過を間接的に保証することができる。

【0049】ステップS 24では、以上のようにして求められたルートコストがリンクテーブルに書き込まれた接続リンクB→Cのルートコスト(初期値=無限大)より小さいかどうか判定する。小さければ、ステップS 25に進み、接続リンクB→Cが経由地リンクB→Cかどうかが判定される。経由地リンクB→Cであれば、経由地点通過フラグ1を設定する(ステップS 26)。

【0050】その後、図1のステップS 28に進み、接続リンクB→Cのルートコストを書き換え、接続リンクB→Cの前リンクとして、リンクA→Bを設定する(ステップS 29)。さらにこの接続リンクB→Cをピボットテーブルに登録し(ステップS 30)、ステップS 5に戻る。ステップS 5では、当該リンクA→Bにつながる他のリンクを探索するが(ステップS 6)、リンクA→Bにつながる他のリンクはこれ以上存在しないので、ステップS 31に進み、ピボットリンクA→Bに処理済フラグを立てる。そしてピボットテーブルを参照し(ステップS 2)、未処理ピボットリンクがあるかどうかを判定する(ステップS 3)。この段階では、リンクB→E、B→Cがピボットテーブルに未処理として登録されている。そこで、未処理ピボットリンクを1つずつ取り出し、上と同じ処理をする。すなわち、接続リンク

を探し出し、そのリンクのルートコストを書き換える。このようにして、リンクE→F, C→A, C→D, E→Fのルートコストが求まっていく。

【0051】なお、リンクC→Aを探索した後、リンクC→Aに接続するリンクをアーチテーブルから拾うと、初期リンクA→Bに戻ってしまう。この場合はステップS 2 4において、リンクA→Bの新しいルートコストを、リンクA→Bのルートコストとして登録されている値(初期値=0)と比較する。新しいルートコストは、経由地を1回通っているので通常のルートコストからaを引いた値である。aは非常に大きな数であるので、

(新しいルートコスト) < (前のルートコスト=0)という関係が成立し、ステップS 2 5-S 3 0に進む。したがって、経路が閉ループとなって探索が無限に続くように見えるが、次に経由地リンクB→Cに到達したときに、経由地点通過済みかどうかの判断がされる(ステップS 2 2)。経由地点通過済みであれば、そこで探索が打ち切られるので無限ループに陥るおそれはない。

【0052】最後に求められた目的地リンクE→Fのルートコストは、先に求められたルートコストに比べて、経由地点通過分aだけ低い値になっている。したがって、ステップS 2 4からS 2 5, 2 7を経てステップS 2 8に進み、ルートコストを書き換える処理をする。さらに、ステップS 2 9において、前リンクをリンクB→EからリンクD→Eに変更し、ステップS 5に戻る。

【0053】そして、ステップS 5, S 3 1, S 2を経由してステップS 3に至り未処理ピボットリンクがあるかどうかを判定する。ピボットリンクがすべて処理済みとなると、ステップS 4 1に行き、リンクテーブルのそれぞれのリンクのルートコストの値に、aを加える。これは、今までルートコストからaを引いたので、元に戻すためである。そして、ステップS 4 2において、目的地リンクであるリンクE→Fの前リンクを辿っていき、経由地点を通過する最適経路を決定する。この最適経路は、A→B, B→C, C→D, C→E, E→Fである。リンクE→Fのルートコストを参照すると、そのルートコストが出発値から目的地までの経由地点を通過する最適経路のルートコストとなる。

(3) 経由地点が複数(N)個ある場合

この場合は、リンクテーブルの経由地点通過フラグの種類をN個とする。

【0054】そして、ステップS 2 2の判断において、他の経由地点のリンクを通過していても当該経由地リンクが未通過であれば、「経由地点未通過」の判断をする。ステップS 2 6の処理における経由地点通過フラグとして、通過した経由地点に応じたフラグを立てる。例えば、通過した接続リンクが経由地点1のリンクであれば、「経由地点通過フラグ1」を立て、接続リンクが経由地点2のリンクであれば、「経由地点通過フラグ2」を立てる。

【0055】このようにすることによって、各経由地点を1回ずつ通る最適経路を見つけることができるようになる。ただし、この発明では、各経由地点を通る順番は特定することができない。さらに、経由地点が複数

(N)個ある場合は、前リンクの欄を 2^N 個とする。例えば経由地点数を2とすれば、通過した経由地点のパターンに応じて $2^2 = 4$ 通りの前リンク(経由地点1非通過、経由地点2非通過の場合; 経由地点1非通過、経由地点2通過の場合; 経由地点1通過、経由地点2非通過の場合; 経由地点1通過、経由地点2通過の場合)を設定する。

【0056】このようにするのは、前に説明したように経由地点が設定されている場合、経路の一部が重なる、すなわち各経由地点を通過するために同じ道を2回以上通ることがあるからである。このとき、通過した経由地点のパターンによって、前リンクの意味が異なってくる。したがって、通過した経由地点のパターンに応じて 2^N 個の欄を設け、それぞれに前リンクを記入するようすれば、当該リンクが1回目の経路を構成する場合と、2回目以後の経路を構成する場合とで経路を区別することができるようになる。

【0057】

【発明の効果】以上のように、本発明の経路計算方法によれば、経由地点が1つ又は複数設定されている場合であっても、各経由地点を1回ずつ通る最適経路を1度の計算で求めることができる。したがって、出発地点から経由地点までの最適経路を計算し、次に経由地点から目的地までの最適経路を計算し、最後に両経路をつなぎ合わせるという方法と比較して、計算処理時間を短くすることができる。

【図面の簡単な説明】

【図1】経由地点の指定がある場合の最適経路計算の方法を説明するフローチャート(図9の続き)である。

【図2】本発明の経路計算方法を実行するナビゲーション装置を示すブロック図である。

【図3】十字路における退出リンクの例を示す図である。

【図4】十字路における進入リンクの例を示す図である。

【図5】コントローラの詳細構成図である。

【図6】ルートコストを説明するための簡単なリンク構成図である。

【図7】出発地点と目的地とを含む経路計算の対象となる範囲内のリンク構成図である。

【図8】接続リンクを次々と求めることによって得られたリンクのツリーを示す図である。

【図9】最適経路計算の方法を説明するフローチャートである。

【図10】経由地点の指定がない場合の最適経路計算の方法を説明するフローチャート(図9の続き)である。

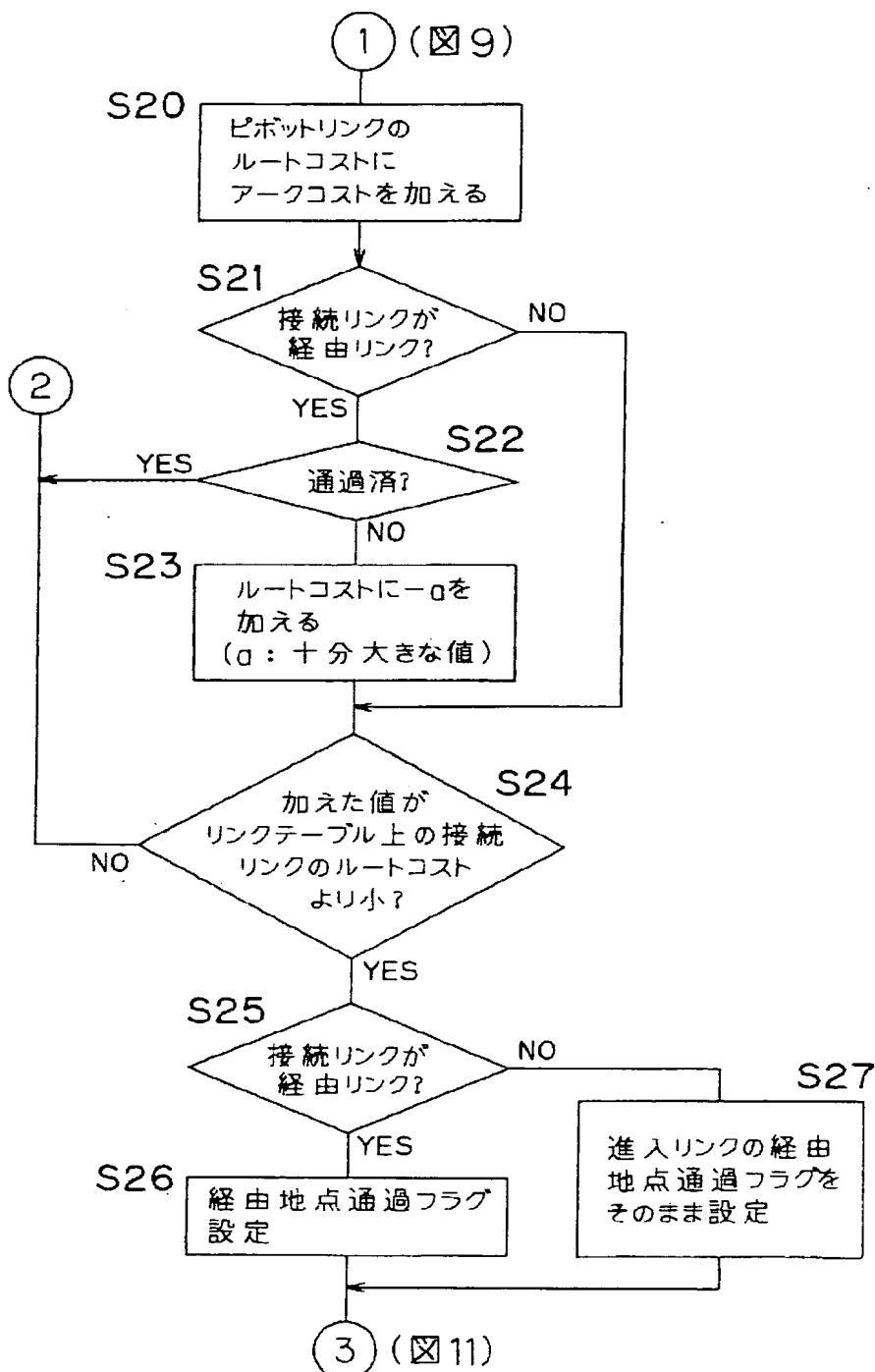
【図11】経由地点の指定がある場合の最適経路計算の方法を説明するフローチャート(図1の続き)である。

【符号の説明】

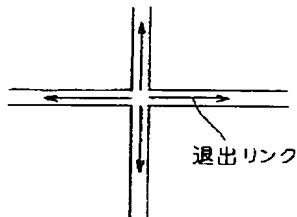
14B 経路ネットワークメモリ

17 コントローラ
26 経路計算処理部
26a CPU
26c 主メモリ

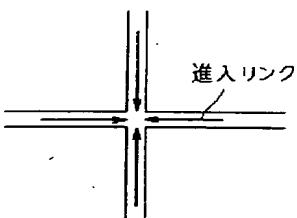
【図1】



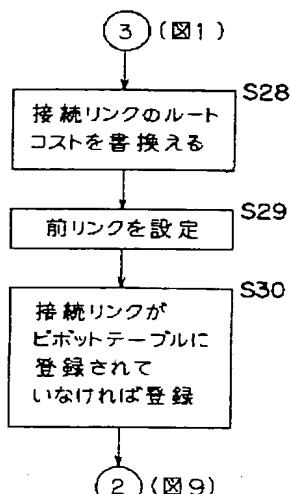
【図3】



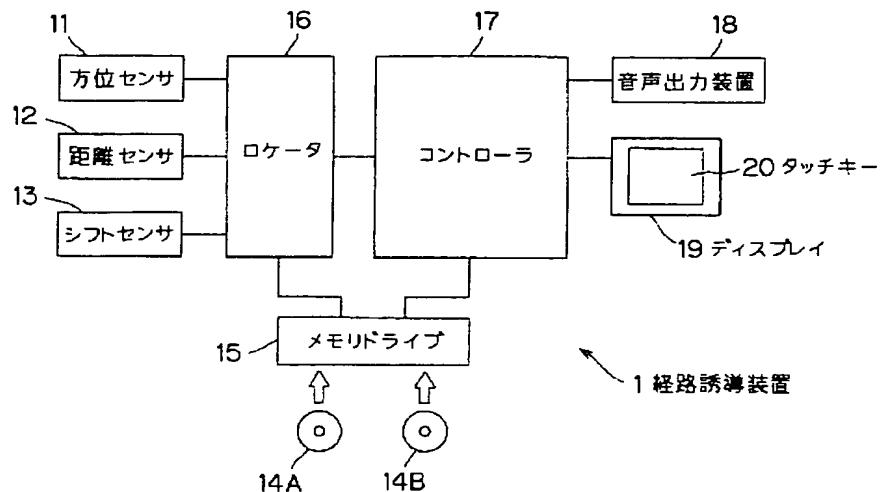
【図4】



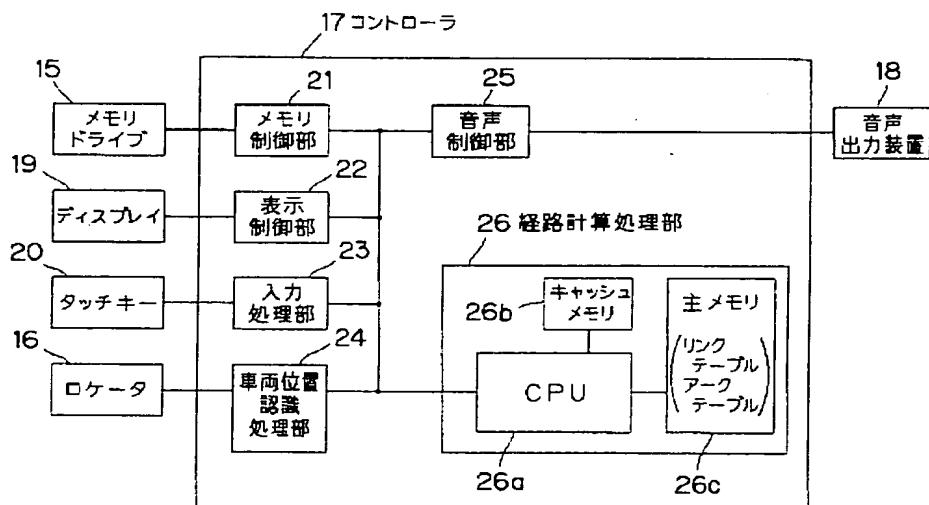
【図11】



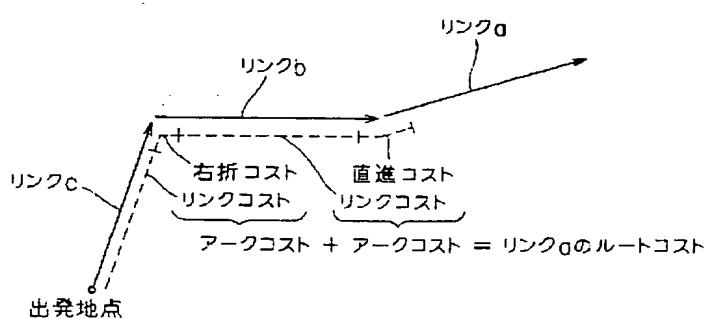
【図2】



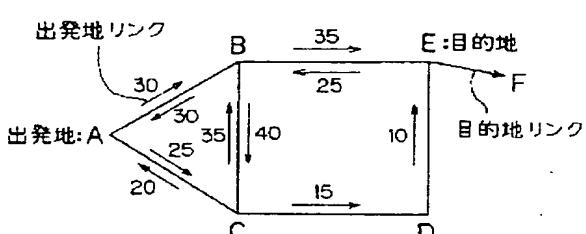
【図5】



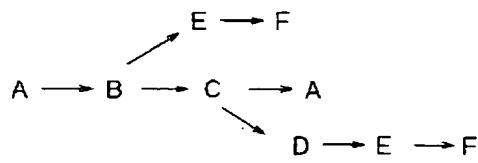
【図6】



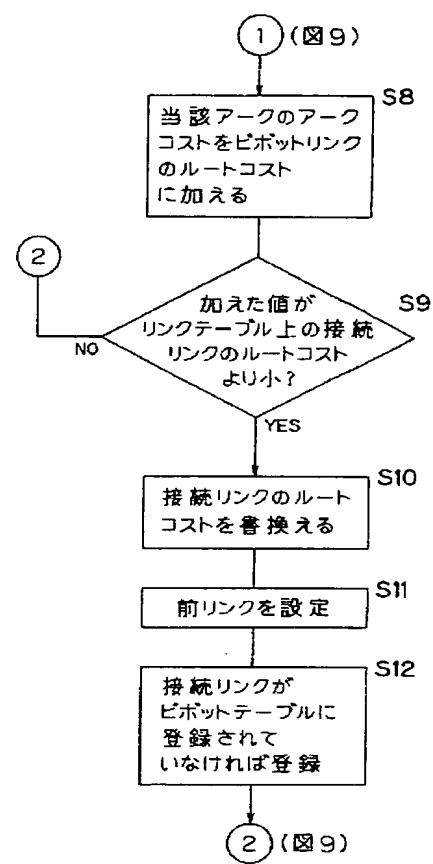
【図7】



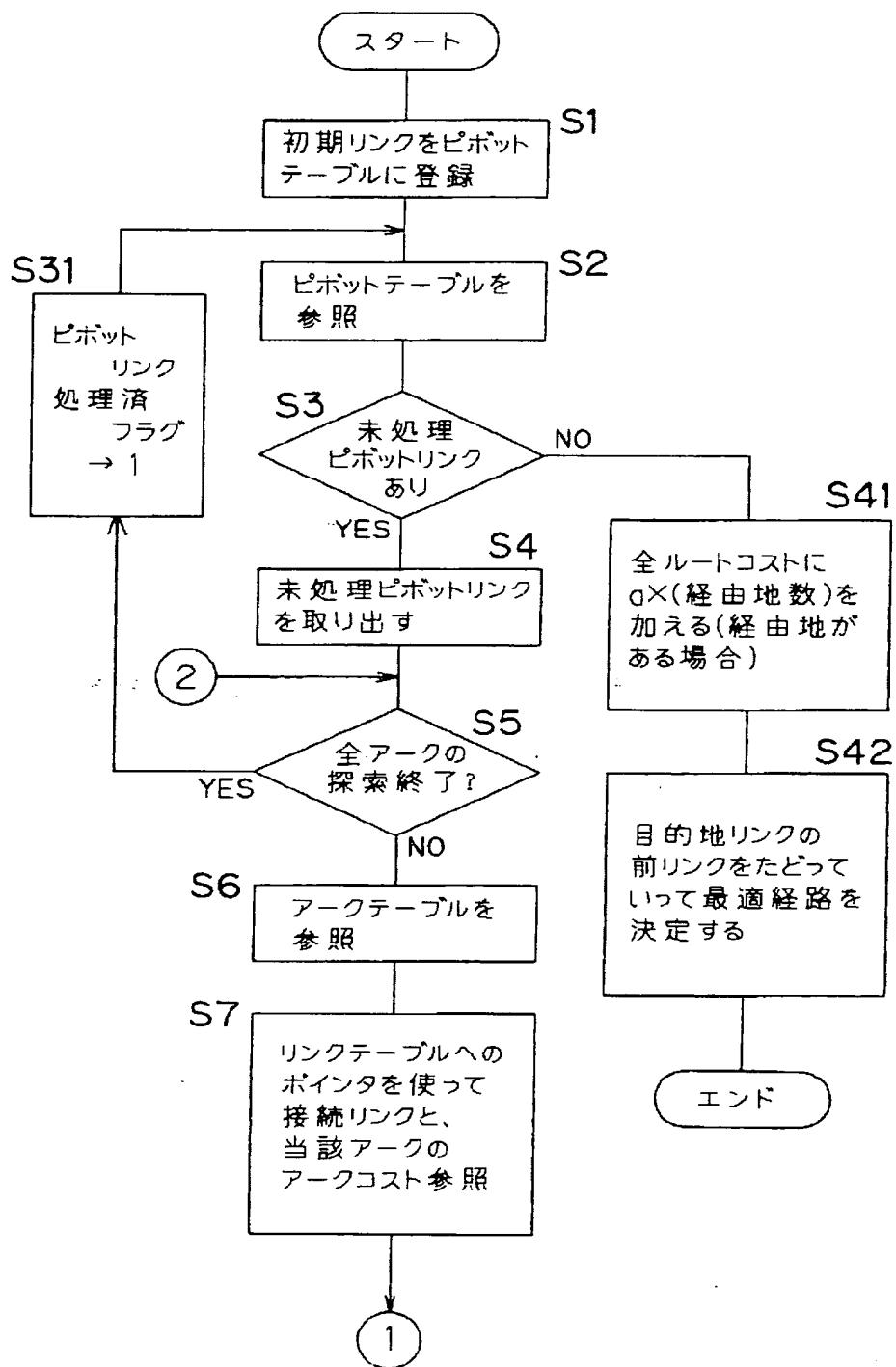
【図8】



【図10】



【図9】



フロントページの続き

(51) Int. Cl. 6

G 06 T 7/60

G 08 G 1/0969

識別記号

府内整理番号

F I

技術表示箇所

7531-3H